

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TREND MICRO, INC.
Petitioner

v.

TAASERA LICENSING LLC

Case No. IPR2023-00799

**PETITION FOR INTER PARTES REVIEW
OF U.S. PATENT NO. 9,092,616**

Filed on behalf of Trend Micro, Inc.

By: Robert M. Hansen

Matthew J. Anderson

Scott B. Amankwatia

The Marbury Law Group, PLLC

11800 Sunrise Valley Drive

15th Floor

Reston, VA 20191

Tel: (703) 391-2900

Fax: (703) 391-2901

TABLE OF CONTENTS

I. INTRODUCTION	1
II. Grounds for Standing (37 C.F.R. § 42.104(a)).....	1
III. Identification of Challenge (37 C.F.R. § 42.104(b))	2
A. Citation of Prior Art	2
B. Statutory Grounds for Challenge (37 C.F.R. § 42.104(b))	3
IV. The '616 patent	3
A. Claim of Priority of the '616 patent	3
B. Overview of the '616 patent.....	3
C. Summary of the Prosecution History	8
D. Level of Ordinary Skill in the Art	10
V. HOW THE CLAIMS ARE UNPATENTABLE	11
A. Overview of Asserted Prior Art	11
i. Mukherjee.....	12
ii. Miliefsky.....	16
iii. Guo	18
B. GROUND 1: Claims 1-6 are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005) and Miliefsky (EX1006).	19
1. Claim 1	23
1a. A method of providing an attestation service for providing runtime operational integrity of a system using a computing platform comprising a network trust agent, an endpoint trust agent, and a trust orchestration server, the method comprising:	23

1b.	sending, by the endpoint trust agent on a monitored device, a dynamic context including endpoint events and actions of the monitored device and applications executing on the monitored device at runtime;.....	33
1c.	receiving, at the trust orchestration server, the dynamic context including the endpoint events of the monitored device and the applications executing on the monitored device at runtime;	36
1d.	analyzing, by the trust orchestration server, the received endpoint events;.....	37
1e.	receiving, by the trust orchestration server, third party network endpoint assessments;.....	38
1f.	generating, by the trust orchestration server, temporal events based at least in part on analyzing the third party network endpoint assessments;	43
1g.	correlating, by the trust orchestration server, the received endpoint events and the generated temporal events; and	45
1h.	generating, by the trust orchestration server, an integrity profile for the system.	47
2.	Claim 2	48
	The method of claim 1, wherein the third party network endpoint assessments are received from an endpoint assessment service.	48
3.	Claim 3	49
	The method of claim 2, further comprising: receiving, by the trust orchestration server, integrity measurement and verification reports from the endpoint assessment service.....	49
4.	Claim 4	51
	The method of claim 1, wherein the third party network endpoint assessments are received from one or more collaboration services.	51

5. Claim 5	52
The method of claim 4, further comprising: receiving, by the trust orchestration server, integrity measurement and verification reports from the one or more collaboration services.....	
6. Claim 6	53
6a. The method of claim 1, wherein the endpoint events of the monitored device are received by a system event correlator of the trust orchestration server from endpoint trust agents, and	
6b. the temporal events are generated by a trust broker of the trust orchestration server and are received by the system event correlator from the trust broker.....	55
C. GROUND 2: Claim 7 is unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005), Miliefsky (EX1006), and Guo (EX1011)....	56
1. Claim 7	59
The method of claim 1, further comprising: performing, by a malware analyzer, detection of code obfuscation and evasion techniques in reverse engineered code of the applications, the code obfuscation and evasion techniques including at least one of: multiple packing, anti-debug, anti-trace, anti-memory, and anti-emulation.	
VI. NO BASIS FOR DISCRETIONARY DENIAL	61
VII. CONCLUSION	63
VIII. Mandatory Notices (37 C.F.R. § 42.8(a)(1))	63

TABLE OF AUTHORITIES

Cases

<i>Advanced Bionics, LLC v. MED-EL Elektromedizinische Gerate GmbH,</i> IPR2019-01469, Paper 6 (Feb. 13, 2020).....	62
<i>Cellco Partnership v. Huawei Device Co.,</i> IPR2020-01117, Paper 10 (Feb. 3, 2021).....	63
<i>Graham v. John Deere Co.</i> , 383 U.S. 1 (1966)	19, 56
<i>KSR Int’l Co. v. Teleflex, Inc.</i> , 550 U.S. 398 (2007)	19, 20, 56
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005) (en banc)	10
<i>Wellman, Inc. v. Eastman Chem. Co.</i> , 642 F.3d 1355 (Fed. Cir. 2011)	11

Statutes

35 U.S.C. § 102(a)/(b)(pre-AIA)	2
35 U.S.C. § 102(e)(pre-AIA)	2
35 U.S.C. § 103(a)(pre-AIA).....	<i>passim</i>

Rules

37 C.F.R. § 42.10(a).....	64
37 C.F.R. § 42.104(a).....	1
37 C.F.R. § 42.8(a)(1)	1, 63
37 C.F.R. § 42.8(b)(2).....	63
37 C.F.R. § 42.8(b)(3).....	64

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 9,092,616 to Kumar <i>et al.</i>
1002	Prosecution History of Kumar <i>et al.</i>
1003	Declaration of Seth James Nielson, Ph.D. under 37 C.F.R. § 1.68.
1004	<i>Curriculum Vitae</i> of A.L. Seth Nielson, Ph.D.
1005	Biswanath Mukherjee, <i>Network Intrusion Detection</i>, IEEE Network, (June 1994).
1006	U.S. Published Patent Application No. 2007/0192867 to Miliefsky.
1007	William R. Cheswick, <i>Firewalls and Internet Security Second Edition, Repelling the Wily Hacker 2</i>, (2003).
1008	Ross Anderson, <i>Network Attack and Defense, Security Engineering: A Guide to Building Dependable Distributed Systems 2</i>, 633-678, (2008).
1009	Andrew S. Tanenbaum, <i>Modern Operating Systems 3</i>, Pearson Prentice Hall, (2008).
1010	U.S. Patent Publication No. 2008/0175266 to Alperovitch <i>et al.</i>
1011	U.S. Patent No. 8,104,089 to Guo <i>et al.</i>
1012	Aubrey-Derrick Schmidt, <i>Static Analysis of Executables for Collaborative Malware Detection on Android</i>, IEEE Xplore, (2009).
1013	U.S. Patent Publication No. 2009/0328222 to Helman
1014	U.S. Patent Publication No. 2012/0072968 to Wysopal
1015	U.S. Patent Publication No. 2012/0110174 to Wotton
1016	J. Andrew Schafer, <i>Windows Management Instrumentation: Create WMI Providers to Notify Applications of System Events</i>, msdn magazine, (October, 2019).
1017	U.S. Patent Publication No. 2008/0244748 to Neystadt <i>et al.</i>
1018	Ilsun You, <i>Malware Obfuscation Techniques: A Brief Survey</i>, IEE Computer Society, 297-300, (2010).
1019	Mamoun Alazab, <i>Towards Understanding Malware Behaviour by the Extraction of API calls</i>, IEE Computer Society, 52-59, (July, 2010).
1020	Request for Comments (RFC) 5209, <i>Network Endpoint Assessment (NEA): Overview and Requirements</i>, (June, 2008).

I. INTRODUCTION

Trend Micro, Inc. (“Petitioner”) petitions for *inter partes* review of claims 1-7 of U.S. Patent No. 9,092,616 to Kumar *et al.*, entitled “SYSTEMS AND METHODS FOR IDENTIFICATION AND REMEDIATION,” (“the ’616 patent,” EX1001), which according to USPTO records is assigned to Taasera Licensing LLC (“Taasera”).

In fashion, often what’s old is new; however, this is not true for patents. In the case of the ’616 patent, the relevant prior art was old enough that the Examiner didn’t find it during prosecution. As will be shown, the basic concepts and details of the ’616 patent claims were public information long before its earliest priority date. Many years later, Patent Owner Taasera sought to patent essentially the same technology. EX1006, ¶¶[0103]-[0104],[0131].

II. Grounds for Standing (37 C.F.R. § 42.104(a))

Petitioner certifies that the ’616 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in this Petition.

III. Identification of Challenge (37 C.F.R. § 42.104(b))

A. Citation of Prior Art

Petitioner cites the following prior art references:

“Network Intrusion Detection” by Biswanath Mukherjee, L. Todd Heberlein, and Karl n. Levitt (“Mukherjee,” EX1005) is a paper that published in 1994.

Mukherjee is prior art under pre-AIA § 102(a)/(b) because it was published and publicly available no later than May 31, 1994, more than 1 year before the priority date of the '616 patent. EX1005, pp.40-41. Mukherjee was not identified or considered during prosecution of the '616 patent.

U.S. Published Patent Application 2007/0192867 to Miliefsky. (“Miliefsky” EX1006) is prior art under pre-AIA § 102(a)/(b) because it published on August 16, 2007, more than one year before the priority date of the '616 patent. EX1006, p.1. Miliefsky was not identified or considered during prosecution of the '616 patent.

U.S. Patent 8104089 to Guo, et al. (“Guo”, EX1011) is prior art under pre-AIA § 102(e) because it was filed on December 31, 2007, several years before the priority date of the '616 patent. EX1011, p.1. Guo was not identified or considered during prosecution of the '616 patent.

B. Statutory Grounds for Challenge (37 C.F.R. § 42.104(b))

Petitioner requests review of '616 patent claims 1-7 (the “Challenged Claims”) on the following grounds:

GROUND 1: Claims 1-6 are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005) and Miliefsky (EX1006).

GROUND 2: Claim 7 is unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005), Miliefsky (EX1006), and Guo (EX1011).

IV. The '616 patent

A. Claim of Priority of the '616 patent

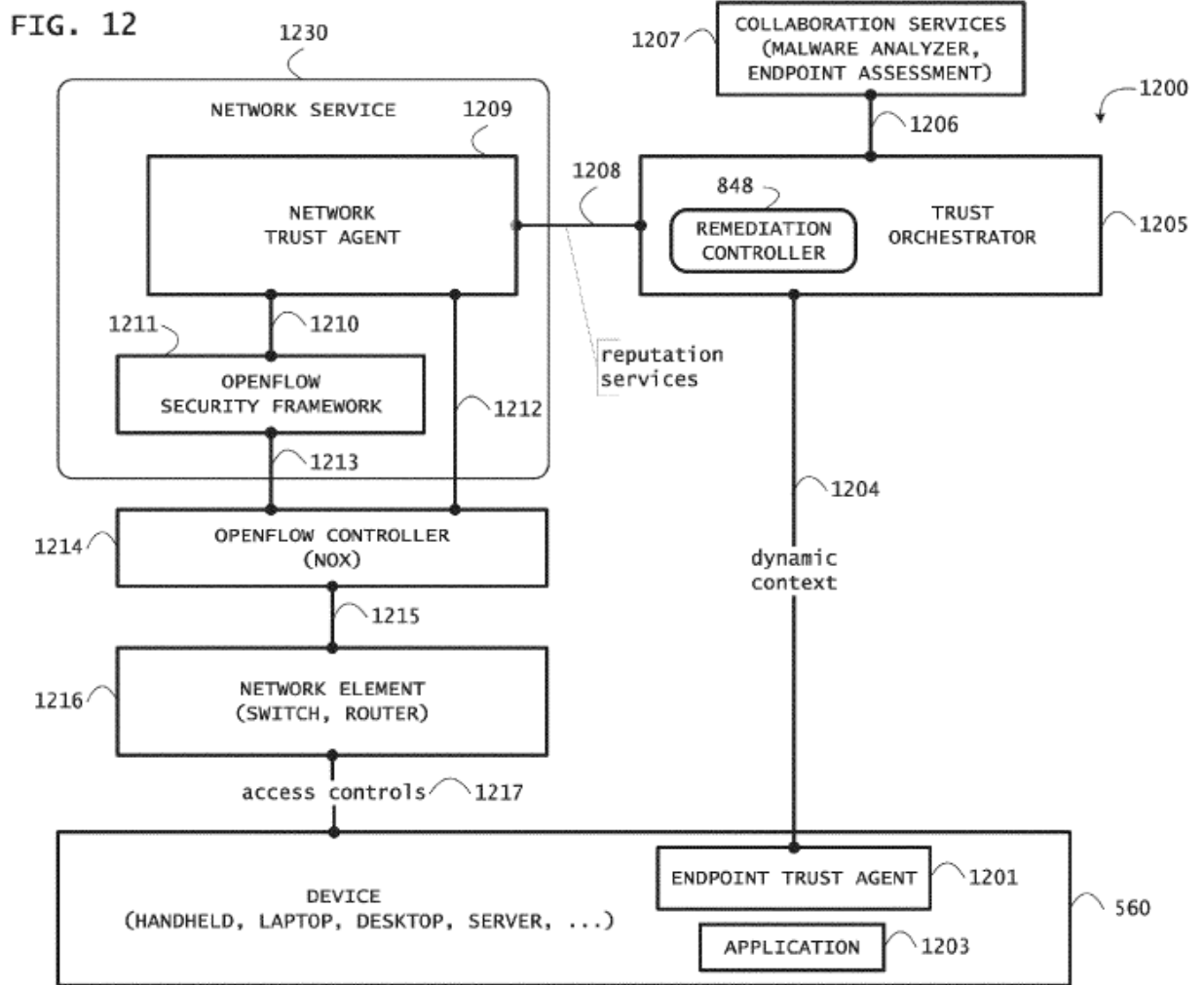
The '616 patent issued on July 28, 2015, from U.S. Patent Application No. 13/559,766, filed July 27, 2012, which claimed priority to U.S. Provisional Application No. 61/641,007, filed May 1, 2012. EX1001, p.1. Solely for the purposes of this Petition, the earliest priority date for the '616 patent is taken to be May 1, 2012 (the “priority date”), so the '616 patent should be examined under pre-AIA statutes.

B. Overview of the '616 patent

The '616 patent is in the field of computer security. EX1001, 1:14-18. As background, the patent recognizes that “[t]he proliferation of applications (business, social networking and gaming software) in Enterprise and cloud

computing ecosystems has significantly increased the attack surface and window of exposure,” and “[t]he proliferation of unmanaged devices such as mobile computing devices (i.e., tablets and smartphones) and Bring Your Own Device (BYOD) policies in Enterprise network environments has increased the risk of advanced coordinated threats.” *Id.*, 2:14-23.

To address this problem, the ’616 patent discloses “systems and methods for providing dynamic operational integrity attestation of application security and a user reputation at runtime.” Attestation systems are components of computer systems that permit reliable statements of evidence about those systems (i.e., that the target system will not engage in defined classes of misbehaviors) to be conveyed to remote parties, including other computers. *Id.*, 3:49-55. FIG. 12 illustrates a system of the ’616 patent. EX1001, 25:9-10; EX1003, ¶73.



EX1001, FIG. 12.

Referring to FIG. 12, the system includes an endpoint trust agent 1201 on a device 560 that may execute an application 1203, a trust orchestrator 1205, a plurality of collaboration services 1207 that may include malware analyzers, network endpoint assessment services, and reputation services, and a network service 1230 that may include a network trust agent 1209. EX1001, 25:15-26.

The endpoint trust agent resides on a device 560, which may be a desktop, server, laptop, or a mobile computing device such as a smartphone or tablet.

EX1001, 13:20-23; 25:65-26:5. The endpoint trust agent monitors operations performed on the device 560 at runtime, including running applications and device functions (e.g., microphone, camera, keyboard, or display functions), and dispatches endpoint events and security service requests to the trust orchestrator. EX1001, 25:65-26:5. Endpoint events pertain to a monitored endpoint device and applications executing thereon. EX1001, 25:65-26:5; EX1002, p.223. The endpoint trust agent is risk and threat agnostic and merely generates events based on the evaluation of rulesets and assessment of runtime behavior based on a threat correlation matrix. EX1001, 26:14-19. The endpoint trust agent sends a dynamic context 1204 that may include the local runtime execution context of an application to the trust orchestrator 1205. EX1001, 25:27-35.

The assessment of risk, and classification and identification of threats is delegated to the trust orchestrator. EX1001, 26:14-19. The trust orchestrator performs a calculus of risk on a global security context, that may include endpoint assessment reports 1206, and sends a system warning 1208 (endpoint threat intelligence) as a subscription-based reputation service to the network trust agent 1209 for network flow remediation. EX1001, 25:27-35. “Based on at least the analysis of the network endpoint assessments, the trust orchestration server generates temporal event (time-based events), which provides assertions of operational parameters of the monitored system.” EX1001, 11:50-65; EX1002,

p.222. The trust orchestrator 1205 sends to the network trust agent 1209 a system warning 1208 for the network device 560. EX1001, 33:7-15.

The trust orchestrator may receive endpoint assessment reports 1206 from a plurality of collaboration services 1207 that may include malware analyzers, network endpoint assessment services, and reputation services. EX1001, 25:18-20, 30-33. Endpoint assessment services perform scheduled policy-based scans to detect known exposures (vulnerability, configuration, compliance and patch level exploits). *Id.*, 20:26-30. A malware analyzer reports capabilities detected through static and dynamic analysis of an image (i.e., device image) and where obfuscated code may be reversed (i.e., reverse engineered) for analysis. EX1003, ¶¶76. “Such image analysis can look for code obfuscation techniques such as anti-debug, anti-trace, anti-memory, and anti-emulation, use of network communication protocols like SMTP, FTP or HTTP, use of embedded uniform resource locators (URLs) or IP addresses, use of encryption methods such as use of the secure sockets layer (SSL) protocol, listening on local ports, remote ports accessed, file system operations, registry operations, operations in temporary, system or non-program folders, memory peek or poke operations, and remote thread creation.” EX1001, 20:12-23; EX1003, ¶¶74-75.

The trust orchestrator generates an integrity profile for an endpoint. The integrity profile represents an aggregation of system warnings (threats such as

malware) identified based on the received temporal and endpoint events. EX1001, 12:41-44. The network trust agent 1209 supports network trustworthiness, for example, by sending messages 1210 to a computer network protocol security framework 1211 middleware to send directives 1213 to a computer network protocol controller 1214 or sending directives 1212 to the computer network protocol controller 1214, to notify by means of a protocol 1215 a computer network protocol enabled network element 1216 to apply access controls 1217 to block or divert traffic flows. EX1001, 25:36-45.

C. Summary of the Prosecution History

Prosecution began with a Non-Final Office Action on July 18, 2013, that rejected all claims as being anticipated or unpatentable over 2009/0328222 to Helman (EX1013), USP 8,214,497 to Alperovitch (EX1010), “Static Analysis of Executables for Collaborative malware Detection on Android” (“Schmidt”)(EX1012), Pat. Pub. 2012/0072968 to Wysopal (EX1014), and/or “Windows Management Instrumentation: Create WMI Providers to Notify Applications of System Events” (“msdn”) (EX1016). EX1004, pp.276-304.

On October 1, 2013, Taasera responded by amending the claims and arguing differences between the claim language and the prior art. *Id.*, pp.201-232. The applicant characterized the claimed invention as follows:

The premise of claim 1 pertains to a solution for detecting infected systems based on (1) observed malicious behaviors *of applications* (e.g., malware or compromised non-malware) and indicators of compromise *at a monitored endpoint device* (e.g. laptop, desktop, server, mobile devices, etc.) and (2) identified stages in the life cycle observed through network dialogs with internal and external entities over the network. The claimed applications are programs executing at the monitored endpoint device; *not* applications on a networked device along a data path that processes single flows on received data. An integrity profile for the system is then generated (on a basis of the endpoint events of the monitored device and applications executing thereon) in order to, e.g., initiate corrective actions to remediate an infected device (e.g., tiered at machine, flow and transactional levels). Thus, any corrective action would be based on endpoint device state and *not* flow based detection of threats in progress. Further, any corrective action would not be triggered by a network flow, intrusion or attack alert. Rather, it would be based on a runtime integrity of the applications executing on the endpoint device, including applications that may not be communicating over the network.”

EX1002, pp.222-23 (emphases original).

A Final Office Action dated January 17, 2014, rejected the claims as unpatentable over Helman, Alperovitch, Wootton, Schmidt, msdn, and/or U.S. Pat. Pub. 2008/0244748 to Neystadt (EX1017). *Id.*, pp.135-172.

After an Examiner-Initiated Interview, Taasera responded by amending the claims to recite, in part “sending, by the endpoint trust agent on a monitored device, a dynamic context including endpoint events and actions of the monitored

device and applications executing on the monitored device at runtime.” *Id.*, pp.91-

94 (underlines in original). Taasera argued that the cited references did not disclose such claim limitation. EX1003, ¶81; EX1001, pp.91-94.

Thereafter, a Notice of Allowance issued on May 22, 2015 with an Examiner’s statement of reasons for allowance that “applicant’s arguments ... are persuasive, as such the reasons for allowance are in all probability evident from the record....” *Id.*, p.33-39.

D. Level of Ordinary Skill in the Art

As established by the Declaration of Dr. Seth Nielson dated April 5, 2023, a person having ordinary skill in the art (PHOSITA) at the time of the earliest claimed priority date for the ’616 patent would have had a bachelor’s degree in computer science, or a related field, or an equivalent technical degree or equivalent work experience, and an additional one to two years of education or experience in the field of computer security topics. More education can supplement practical experience and vice versa. EX1003, ¶83.

E. Claim Construction

In this proceeding, claims are interpreted under the same standard applied by Article III courts (i.e., the *Phillips* standard). 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

So far in parallel litigation neither Taasera nor any defendants have proposed a claim construction. For purposes of this proceeding only, Petitioner proposes that claim terms should be afforded their plain and ordinary meanings and no specific claim constructions are necessary to apply the asserted prior art to the Challenged Claims. ¹ *See Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011) (“[C]laim terms need only be construed to the extent necessary to resolve the controversy.”).

V. HOW THE CLAIMS ARE UNPATENTABLE

The following discussion and the accompanying evidence demonstrate that there is a reasonable likelihood that Petitioner will prevail with respect to the Challenged Claims.

A. Overview of Asserted Prior Art

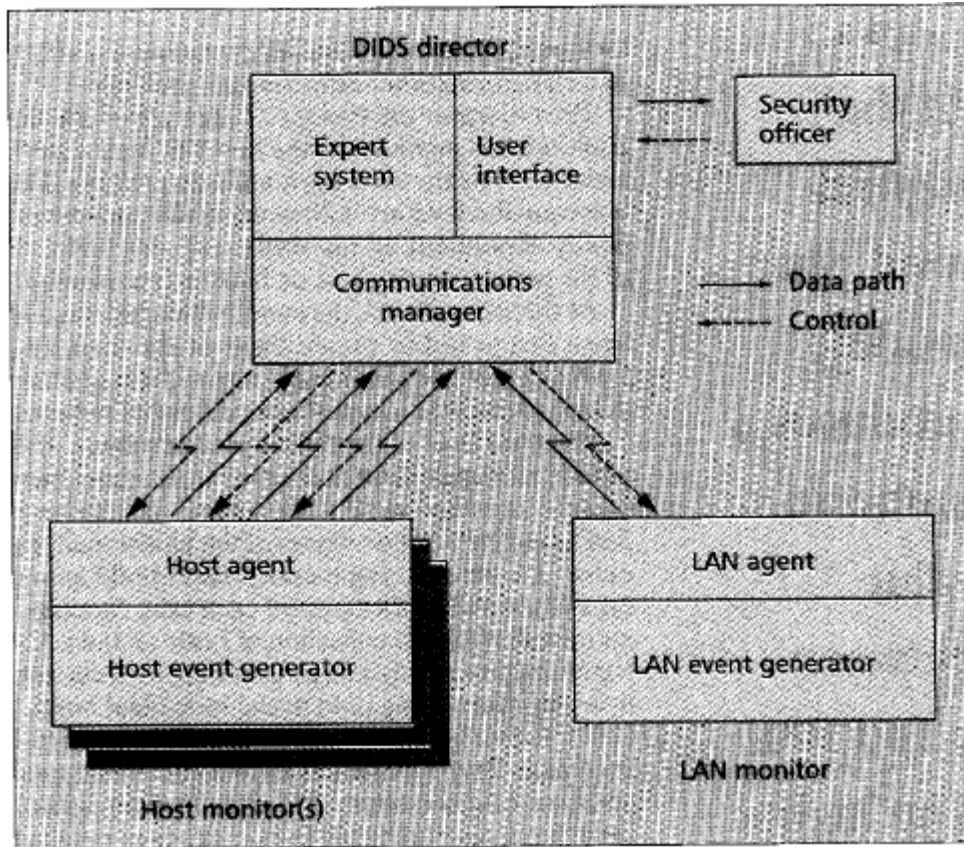
Petitioner submits that claims 1-6 of the '616 patent are invalid as unpatentable under 35 U.S.C. § 103(a) in view of the combination of Mukherjee and Miliefsky.

¹ Petitioner reserves the right in other proceedings to construe the Challenged Claims and pursue indefiniteness theories.

i. Mukherjee

Mukherjee is directed towards computer security. EX1005, p.1,8; EX1003, ¶85. Mukherjee describes several host-based and network-based intrusion detection systems (IDSs) that are combined under a single network security monitor (NSM) architecture. *Id.* “Intrusion detection is defined to be the problem of identifying individuals who are using a computer system without authorization (i.e., “crackers”) and those who have legitimate access to the system but are abusing their privileges (i.e., the “insider threat”).” EX1005, p. 2. Mukherjee explains, that IDSs are “based on the beliefs that an intruder’s behavior will be noticeably different from that of a legitimate user and that many unauthorized actions are detectable.” *Id.* Typically, IDSs employ statistical anomaly and rule-based misuse models in order to detect intrusions.” *Id.*

One such system is a Distributed Intrusion Detection and Alerting System (DIDS), the architecture for which is shown in FIG. 2. EX1005, pp.8,11, Fig.-2; EX1003, ¶86-87.



EX1005, FIG. 2.

Among other things, the DIDS system monitors and provides security for one or more host computer devices, which can be hosts or servers from different vendors. EX1005, p.8, Fig.-1; EX1003, ¶¶91. In the DIDS architecture, each host device or computer in the monitored domain is equipped with a host monitor. EX1005, p.12; EX1003, ¶¶89. A single LAN monitor is provided for each broadcast LAN segment in the monitored network. *Id.* The host and LAN monitors are responsible for the collection of evidence of unauthorized or suspicious activity, while a DIDS director is primarily responsible for its evaluation. EX1003, ¶¶89-91. The host and LAN monitors report any “interesting” events to the DIDS director.

EX1005, p.12. The director employs an expert system (rule-based analysis system) to detect possible attacks. EX1005, pp.12-13; EX1003, ¶¶92. The DIDS architecture performs information aggregation so that even if the activity of a network-wide user may not be suspicious on a single host, the aggregate behavior may be suspect or unpermitted. EX1005, p.13; EX1003, ¶¶94. This architecture also performs “accountability” by tying users with their actions. *Id.* This architecture is not restricted as to the network encompassed and protected and may be implemented for internet-based networks. EX1003, ¶¶95.

The host monitor consists of a host event generator (HEG) and a host agent. EX1005, p.12, Fig.-2. The HEG collects and analyzes audit records from the host's operating system (OS) and reports them to DIDS. *Id.*, pp.11-12. The audit records are also scanned for notable events, which are transactions that are of interest independent of any other records. *Id.* These events include failed events, user authentications, changes to the security state of the system, and any network access such as *rlogin* and *rsh*. *Id.*, p.12. These notable events are sent to the DIDS director for further analysis. The HEG also tracks user sessions and reports anomalous behavior aggregated over time through user and group profiles, which is supported by the integration of the HAYSTACK intrusion detection algorithm into DIDS. *Id.*, pp.12-15. The host agent handles all communications between the host monitor and the DIDS director. *Id.*, pp.12.

The LAN monitor includes a LAN event generator (LEG) and a LAN agent.

The LEG may operate as a subset of the NSM that observes traffic on its segment of the LAN in order to monitor host-to-host connections, services used, and volume of traffic. *Id.*, p.12. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns. *Id.*, pp.12-13.

The DIDS director consists of three major components that are all located on the same dedicated workstation - a communications manager, an expert system, and a user interface. *Id.*, p.11. The communications manager is responsible for the transfer of data between the director and each of the host and the LAN monitors. *Id.*, pp.12-13. The expert system uses rule-based logic for evaluating and reporting on the security state of the monitored system. *Id.* The DIDS director receives the reports from the host and the LAN monitors; and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole. *Id.* The DIDS director's user interface allows the SSO interactive access to the entire system.

The expert system of the DIDS director in Mukherjee includes an intrusion detection model (IDM) that is used to analyze network data and other data to efficiently detect intrusions. *Id.*, pp.12-13. This efficiency, in part, is derived from the multiple layers of abstraction that process network information before passing

it onto a higher level. EX1003, ¶¶92-94. In this manner, raw audit records are converted to source agnostic record, which are converted to source-based (subject) records, which are converted to contextualized events, which are converted to threat-based combinations of events, which are converted to risk assessments per threat as a single value as the final, sixth layer/level. EX1005, p.13; EX1003, ¶93. The processing within the IDM as disclosed in Mukherjee is fundamental to threat analysis. EX1003, ¶95.

ii. Miliefsky

Miliefsky is directed toward “computer security, and more particularly to a micro-appliance for use in defending against common vulnerabilities and exploits.” EX1006, ¶¶[0004],[0076]; EX1003, ¶96. To address issues in conventional anti-hacker security systems Miliefsky discloses automatically and proactively identifying and remediating the vulnerabilities used by attackers to exploit a system.

The system and methods described herein include, among other things, security systems that provide proactive automated defense against hackers by automatically finding, reporting, communicating with countermeasures about and removing the common vulnerabilities and exposures (CVEs) that they exploit. Accordingly, the systems described herein provide for proactive security by determining the components that exist on a network system and generating a list of network assets. EX1006, ¶[0039] (emphasis added).

Miliefsky discloses that the system scans existing devices to create an inventory, including identifying “rogue” devices that are vulnerable. EX1003, ¶¶98. The system also accesses public sources of vulnerability information (third party vulnerability data) to identify known problems in the system and build a “true risk profile.” EX1003, ¶¶97-99; EX1006, ¶¶0040]. In the system of Miliefsky, a central processing engine communicates with and manages other engines and access control for nodes of the network based on assessment of those nodes. EX1006, ¶¶0068],[0077],Fig.-7, Fig.-17a. Specifically, Miliefsky states:

An Information Security (INFOSEC) software engine acts as a gateway between users, personal computers, servers, services and the computer network (internet, intranet, extranet, wide area network, wireless network or local area network). The information Security (INFOSEC) Engine controls the computer-based network scanning, standards-based vulnerability assessment through common vulnerabilities and exposures (CVEs) testing, reporting and remediation as well as interfacing with the INFOSEC ENGINE PLUGIN. ... The INFOSEC Engine, at initialization, is able to determine which CVE tests are available as well as which INFOSEC Engine Plugins are available and then to relay this information to the Secure GUI for administration, control and management.

EX1006, ¶¶0049].

The central analysis configuration of the system in Miliefsky allows for intrusion detection of heterogeneous hardware and software environments and may

be combined with agent-based systems and clientless systems with no local host application. *Id.*, ¶¶[0054]-[0057],[0062]-[0067],[0077],[0094]. Miliefsky describes integration of the central analysis engine with countermeasures, vulnerability scanners, network event logs, databases, and various other subsystems. EX1006, ¶¶[0082]-[0095], EX1003, ¶100-104. These integrations enable the central analysis engine to assess devices/nodes as they connect to the network and to summarize a network security state. EX1006, ¶¶[0086],[0094]-[0096]. Thus, Miliefsky proposes an architecture for network security that enables and encourages integration of further analytical tools and devices. EX1006, ¶[0098]; EX1003, ¶106.

iii. Guo

The disclosure of Guo is directed to a memory-based method of analyzing packed application executables as they install to detect obfuscated code in real time by reversing the obfuscation. EX1011, 2:20-47; EX1003, ¶108. The obfuscated code is then scanned after it is unpacked during install or runtime to determine if the obfuscated code is malicious. EX1011, 5:1-16,7:47-65; EX1003, ¶¶108-110. Guo discloses that this method is more efficient than prior manual reverse-engineered unpacking methods or emulation methods for analysis of code before install or runtime. EX1011, 1:43-67; EX1003, ¶109.

Because Guo allows the binary or executable to unpack itself, the disclosed method is agnostic to which packer was used. EX1011, 4:1-35; EX1003, ¶¶110-112. Indeed, even if multiple packers were used on various portions of the executable, Guo discloses monitoring memory events to identify the completion of the unpacking by each packer in order to scan the unpacked code before execution. EX1011, 4:1-35, 7:30-55; EX1003, ¶111. In this manner, Guo is accomplishing automated reverse engineering of the obfuscated code by allowing the unpackers of the executable unpack it themselves. EX1011, 5:45-55; EX1003, ¶115. Once the obfuscated code is unpacked and analyzable, Guo scans it with an anti-virus scanner to identify any malicious code that has been unpacked. EX1003, ¶¶113-114; EX1011, 4:65-5:16. Therefore, the various techniques disclosed in Guo are used to undo the obfuscation of code in executables and scan it for viruses. EX1003, ¶¶110-112.

B. GROUND 1: Claims 1-6 are unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005) and Miliefsky (EX1006).

Petitioner evaluates the scope and content of the prior art, any differences between the art and the claim, and the knowledge of person of ordinary skill in the art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is unaware of any secondary considerations relevant to the '616 patent and a more than adequate written

rationale is provided to support the combination of Mukherjee and Miliefsky under *KSR*. 550 U.S. at 418.

Motivation to Combine Mukherjee and Miliefsky

A PHOSITA would have been motivated to combine the teachings of Mukherjee and Miliefsky because both Mukherjee and Miliefsky disclose computer security systems that detect and ameliorate vulnerabilities. EX1003, ¶117.

Mukherjee discloses “Intrusion detection systems (IDSs) that are based on the belief that an intruder’s behavior will be noticeably different from that of a legitimate user and that many unauthorized actions are detectable.” EX1005, p.1. Mukherjee further states that IDS’s may include “rules based on... known system vulnerabilities” for detecting intrusive behavior. EX1005, p.5; EX1003, ¶117.

Similar to Mukherjee, Miliefsky is also directed toward addressing vulnerabilities because Miliefsky discloses a computer security system that can be used “in defending against common vulnerabilities and exploits.” EX1006, ¶¶[0004],[0098]; EX1003, ¶118.

A PHOSITA would be motivated to combine Mukherjee and Miliefsky to provide compatible systems that detect current threats **and** mitigate future threats. EX1003, ¶¶119-122. Mukherjee uses vulnerabilities to detect threats, and

Miliefsky uses this information to mitigate and proactively block intruders. *Id.*

Specifically, Miliefsky states:

The system and methods described herein include, among other things, security systems that provide proactive automated defense against hackers by automatically finding, reporting, communicating with countermeasures about and removing the common vulnerabilities and exposures (CVEs) that they exploit. Accordingly, the systems described herein provide for proactive security by determining the components that exist on a network system and generating a list of network assets.

EX1006, ¶[0039] (emphasis added).

Mukherjee discloses that IDS systems are designed to **be integrated or used with** systems that provide proactive defenses. EX1003, ¶119-120. Specifically, Mukherjee states:

It is anticipated that a growing set of tools, including incident-handling tools and network-management tools, will be used in conjunction with the intrusion-detection functions of DIDS. This will give the SSO the ability to actively respond to attacks against the system in real time. Incident-handling tools may consist of possible courses of action to take against an attacker, such as cutting off network access, a directed investigation of a particular user, removal of system access, etc. Network- management tools that are able to perform network mapping would also be useful.

EX1005, p.12 (emphasis added).

Miliefsky incorporates IDS systems because Miliefsky discloses that a security appliance includes INFOSEC (Information Security) components and plugins of various types including IDS. EX1003, ¶121. Miliefsky states:

In the event the INFOSEC countermeasure is an IDS, dynamic alerting of the IT manager or an alternative alert recipient and sharing the related CVE tests data with the IDS to help the IDS reduce false positives in the IDS alerting module as well as reduce the traffic load related to intrusion detections which attack a particular IP address that is not susceptible to that particular attack methodology based upon the related CVE tests data and will take place through the Countermeasure Communications Client plug-in which has been written specifically for that MAKE, MODEL and VERSION IDS.

EX1006, ¶[0064] (emphasis added).

Thus, a PHOSITA would recognize that Miliefsky discloses combining Miliefsky's data into an IDS's operation to reduce false positives and reduce traffic load. EX1003, ¶122.

Further, a PHOSITA would have known there was a likelihood of success in combining Mukherjee and Miliefsky because Mukherjee discloses that integration is expected and desired, and Miliefsky discloses that IDS incorporation is expected and desired. Thus, it would be obvious to try the combination with a high expectation of success. EX1003, ¶119-120,123.

1. Claim 1

1a. A method of providing an attestation service for providing runtime operational integrity of a system using a computing platform comprising a network trust agent, an endpoint trust agent, and a trust orchestration server, the method comprising:

Mukherjee discloses “an attestation service for providing runtime operational integrity of a system.” EX1005, p.12; EX1003, ¶¶127-128. Mukherjee discloses a system that can watch activities on hosts, receive and analyze events generated by the host, and based on the same, develop high-level hypotheses about intrusions and the overall security of the monitored environment. EX1003, ¶¶127-130. Mukherjee states:

The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host’s operating system. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. These include, among others, failed events, user authentications, changes to the security state of the system, and any network access such as rlogin and rsh. These notable events are then sent to the director for further analysis.

EX1005, pp.11-12 (emphasis added).

The expert system is responsible for evaluating and reporting on the security state of the monitored system. It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole. The expert system is a rule-based system with

simple learning capabilities. The director's user interface allows the SSO interactive access to the entire system. The SSO can watch activities on each host, watch network traffic (by setting "wiretaps"), and request more specific types of information from the monitors.

EX1005, p.12 (emphasis added).

The expert system uses rules derived from a hierarchical Intrusion Detection Model (IDM). The IDM describes the transformation from the distributed raw audit data to high-level hypotheses about intrusions and about overall security of the monitored environment.

EX1005, p. 12.

At the highest level, the model produces a numeric value between one and 100 which represents the overall security state of the network. The higher the number, the less secure is the network. This value is a function of all the threats for all the subjects on the system.

EX1005, p. 13.

Mukherjee explains that its system provides a user interface for a System Security Officer (SSO) to watch activities on each host and request specific types of information from the monitors. EX1003, ¶128; EX1005, p.12. The SSO is described as the user to the "user interface" of the DIDS system, wherein the "DIDS director consists of three major components that are all located on the same dedicated workstation – a communications manager, an expert system, and a user interface." EX1003, ¶128; EX1005, p.12. A PHOSITA would recognize that

reporting activities dynamically that pertain to the integrity of the system to be “providing runtime operational integrity of [the] system.” EX1003, ¶130.

Mukherjee further discloses monitors that report “interesting” events to a centrally located DIDS director. EX1003, ¶129. Specifically, Mukherjee states:

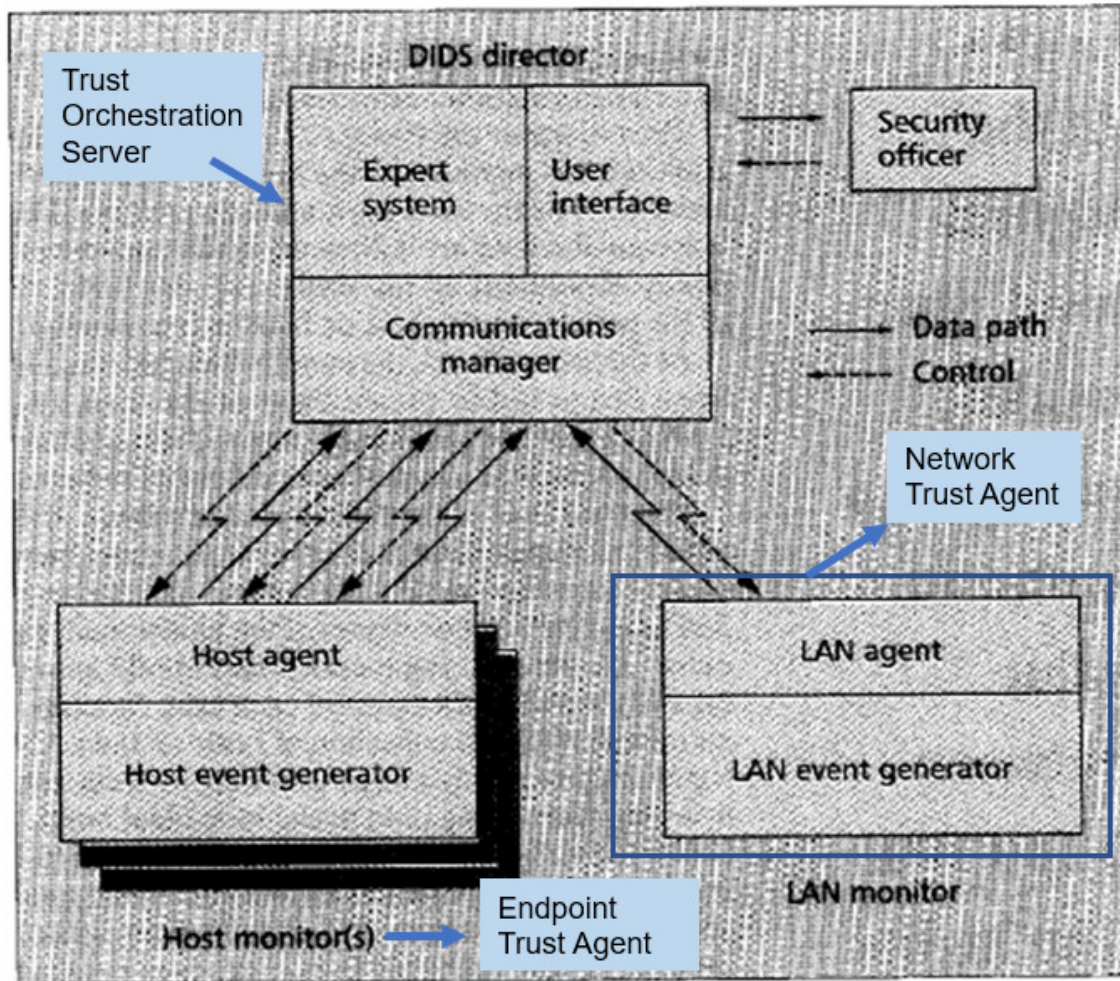
In DIDS, the host and LAN monitors report any “interesting” events, which may possibly lead to intrusive activity, to a centrally-located DIDS director. The director employs an expert system to detect possible attacks. An initial prototype of DIDS has successfully demonstrated its ability to track users as they move around the network (possibly attempting to hide their true identities) and to identify doorknob rattling attacks [32,34]. The DIDS architecture performs information aggregation so that even if the activity of a network-wide user may not be suspicious on a single host, the aggregate behavior may be suspect or unpermitted. One of the strengths of this architecture is that it performs “accountability” by tying users with their actions.

EX1005, p.11, (emphasis added).

Thus, a PHOSITA would understand reporting events which lead to intrusive activity in a computer system and monitoring such activity via the DIDS director to detect possible attacks to correspond to an assessment at runtime or “runtime operational integrity of a system.” EX1003, ¶¶129-130.

Mukherjee discloses that the attestation service is provided “using a computing platform comprising a network trust agent, an endpoint trust agent, and a trust orchestration server.” EX1005, pp.8, 11-16; EX1003, ¶131. Mukherjee

discloses a LAN Monitor (corresponding to “network trust agent”), a Host Monitor (corresponds to an “endpoint trust agent”), and a DIDS director (corresponding to a “trust orchestration server”). EX1003, ¶¶131-132.



EX1005, FIG. 2 (annotated)

Regarding the “network trust agent,” Mukherjee discloses that the LAN monitor supports network trustworthiness by observing and reporting on network events useful in identifying intrusion behavior. EX1003, ¶133. Specifically, Mukherjee states:

In DIDS, the host and LAN monitors report any “interesting” events, which may possibly lead to intrusive activity, to a centrally-located DIDS director...The components of DIDS are the DIDS director, a single host monitor per host, and a single LAN monitor for each broadcast LAN segment in the monitored network.

EX1005, p.11 (emphasis added).

Like the host monitor, the LAN monitor consists of a LAN event generator (LEG) and a LAN agent. ... Its main responsibility is to observe all of the traffic on its segment of the LAN in order to monitor host-to-host connections services used, and volume of traffic. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns.

EX1005, p.12 (emphasis added).

...each layer representing the result of a transformation performed on the data [e.g., from a lower level]. The objects at the first (i.e., lowest) level are the audit records provided by the host OS, by the LAN monitor, or by any third-party auditing package. The objects at this level are both syntactically and semantically dependent on the source. At this level, all of the activity on the host or LAN is represented. At the second level, the event (which has already been discussed in the context of the host and LAN monitor) is both syntactically and semantically independent of the source standard format for events.

EX1005, p.13 (emphasis added).

A PHOSITA would recognize the LAN monitor of Mukherjee to be the same as or equivalent to a “network trust agent,” consistent with the teachings of the ’616 patent, because both are components that support or ensure network trustworthiness. EX1003, ¶¶134-135.

Additionally, to the extent that a “network trust agent” may include control capabilities (which is not conceded), Miliefsky discloses a Countermeasure Communications engine that directly or indirectly controls network flows by controlling trusted gateways to the network (e.g., VPNs). EX1003, ¶¶136-140; EX1006, ¶¶[0055],[0062],Fig.-11. A PHOSITA would have recognized the Countermeasure Communications engine of Miliefsky is the same as or corresponds to a “network trust agent” at least because the Countermeasure Communications engine quarantines untrusted network assets. EX1003, ¶¶136,140.

Further, Mukherjee discloses a “endpoint trust monitor” in the form of a host monitor that observes and sends information about the local execution context of applications, services, and activity at a host device. EX1005, pp.11-12; EX1003, ¶¶142-143. The host monitor includes a host event generator (HEG) that collects audit data from the operating system. EX1005, pp.11-12; EX1003, ¶142. “The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host’s operating system. The audit

records are scanned for notable events.” EX1005, p.11; EX1003, ¶142. The host monitor also observes network activities and resources consumed by network activities (via applications and services) because the host monitor tracks network access (e.g., “collects and analyzes audit records from the host’s operating system”). EX1003, ¶145; EX1005, p.5. The audit records are scanned for notable events, such as any access such as rlogin and rsh. EX1005, pp.11-12. Thus, a PHOSITA would recognize that the host monitor is configured to monitor activities at the host computer. EX1003, ¶146.

The host monitor assesses operational integrity based on rulesets by tracking a behavioral measure of activity on the host. EX1003, ¶145. This integrity assessment includes “[a]bout two dozen “features” (behavioral measures) of the user’s session... [such as] time of work, number of files created, number of pages printed, etc.” EX1005, p.5. This analysis is part of the HAYSTACK algorithm. As Mukherjee explains, “[t]he HEG also tracks user sessions and report [sic] anomalous behavior aggregated over time through user and group profiles and the integration of the HAYSTACK intrusion detection algorithm [5] into DIDS.” EX1005, p.12 (emphasis added). Therefore, a PHOSITA would recognize the host monitor as an “endpoint trust agent” as claimed and described in the ‘616 specification. EX1003, ¶¶141,145,146; EX1001, 13:4-18.

A PHOSITA would have recognized the host monitor of Mukherjee is the same as or corresponds to the “endpoint trust agent” because the host monitor assesses operational integrity based on rulesets, observes the operation of other applications on the system and their network activities, and uses resources utilization monitors in order to assess security. EX1003, ¶141-146.

Further, regarding the “trust orchestration server,” the ’616 patent describes that “the trust orchestrator 430 comprises a system event correlator 410 configured to use a risk correlation matrix 411 to generate an integrity profile 420 for the monitored system (i.e., the target system),” where for the risk correlation matrix “each cell is a set of alerts triggered through an event correlation matrix 633 and rulesets 631 (rule expressions).” EX1001, 15:33-44; EX1003, ¶147-148.

Mukherjee discloses a “trust orchestration server” in the form of the “DIDS director”. EX1003, ¶149. Mukherjee states:

The DIDS director consists of three major components... a communications manager, an expert system, and a user interface... The expert system is responsible for evaluating and reporting on the security state of the monitored system. **It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole.**

EX1005, p.12 (emphasis added).

The DIDS director includes an expert system of rules based logic that correlates system events. EX1003, ¶150; EX1005, p.13. The different levels of the IDM are part of the expert system because the “DIDS utilizes a rule-based (or

production) expert system written in CLIPS. The expert system uses rules derived from a hierarchical Intrusion Detection Model (IDM).” EX1005, p.12-13; EX1003, ¶150. Mukherjee discloses correlating events in the “fourth layer” of the Intrusion Detection Model:

The fourth layer of the model introduces the event in context. There are two kinds of context: temporal and spatial. As an example of temporal context, behavior which is unremarkable during standard working hours may be suspicious during off-hours [35]. The IDM, therefore, allows for the application of information about wall-clock time to the events it is considering. Wall-clock time refers to information about the time of day, weekdays versus weekends and holidays, as well as periods when an increase in activity is expected. In addition to the consideration of external temporal context, the expert system uses time windows to correlate events occurring in temporal proximity. This notion of temporal proximity implements the heuristic that a call to the UNIX who command followed closely by a login or logout is more likely to be related to an intrusion than either of those events occurring alone. Such related patterns of behavior are also referred to as attack signatures [23,24].”

EX1005, p.13.

A PHOSITA would have recognized the “DIDS director” of Mukherjee and its IDM logic as corresponding to the “trust orchestration server” at least because the “DIDS director” includes a system event correlator. EX1003, ¶¶150-151.

Thus, a PHOSITA would have recognized the “DIDS director” to be the same thing as or correspond to a “trust orchestration server.” EX1003, ¶151.

A PHOSITA would have been motivated to incorporate Miliefsky’s countermeasure communications engine into Mukherjee’s IDS in order to prevent malicious actions on the system in real time in response to Mukherjee’s detected abnormalities in the network and on the host. EX1003, ¶153. Mukherjee explains that:

It is anticipated that a growing set of tools, including incident-handling tools and network-management tools, will be used in conjunction with the intrusion-detection functions of DIDS. This will give the SSO the ability to actively respond to attacks against the system in real time. Incident-handling tools may consist of possible courses of action to take against an attacker, such as cutting off network access, a directed investigation of a particular user, removal of system access, etc. Network-management tools that are able to perform network mapping would also be useful.

EX1005, p.12 (emphasis added).

Thus, a PHOSITA would have been motivated to combine Mukherjee and Miliefsky such that Mukherjee’s capabilities can actively respond to attacks. EX1003, ¶154. A PHOSITA would have been motivated to specifically use Miliefsky (vs. other prior art) because Miliefsky describes plugging its device into IDS systems (Mukherjee’s DIDS systems). *Id.* As described in the general motivation to combine above, Miliefsky specifically describes a plug-in based architecture to combine with Mukherjee’s DIDS systems. EX1003, ¶¶117-123,154.

Thus, it would have been obvious to try Miliefsky's disclosures with Mukherjee's with minimal experimentation and a high chance of success. *Id.*

For these reasons, a PHOSITA would understand that the combination of Mukherjee and Miliefsky discloses the limitations recited in claim limitation 1[a]. EX1003, ¶156.

1b. sending, by the endpoint trust agent on a monitored device, a dynamic context including endpoint events and actions of the monitored device and applications executing on the monitored device at runtime;

Mukherjee discloses the host monitor ("endpoint trust agent") is "on a monitored device" because it describes that each host device or computer ("monitored device") in the monitored domain is equipped with a host monitor ("endpoint trust agent"). EX1005, p.11; EX1003, ¶159. Mukherjee describes that a host device has a "host operating system (OS)," and as such, is a computer device. EX1005, pp.2,12; EX1003, ¶159; EX1009.

Mukherjee discloses "a dynamic context including endpoint events and actions" in that it describes monitoring events, authentications, and state changes on the host. EX1005, p.12; EX1003, ¶159. Specifically, Mukherjee states:

The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host's operating system. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. *These include, among others, failed events, user authentications, changes to the security state of the system, and any*

network access such as rlogin and rsh. These notable events are then sent to the director for further analysis. The HEG also tracks user sessions and report anomalous behavior aggregated over time through user and group profiles and the integration of the HAYSTACK intrusion detection algorithm [5] into DIDS. The host agent handles all communications between the host monitor and the DIDS director.

.

EX1005, p.12 (emphasis added).

Since these events provided as examples above are runtime events, a PHOSITA would understand that they constitute dynamic context. EX1003, ¶160; EX1001, 25:27-35. The collector of this information is the host event generator which is a part of the host monitor on the host (corresponding to “endpoint trust agent on a monitored device”). EX1005, pp.2,12; EX1003, ¶159. Thus, Mukherjee teaches collecting these events and actions (corresponding to “dynamic context”) on the host and collecting events and actions by the host (e.g., network access) (corresponding to “endpoint events and actions of the monitored device”). EX1003, ¶¶159-160.

Additionally, the notable events from the host are sent to the DIDS director while the host is in operation as stated in the above quote (corresponding to “sending...a dynamic context including endpoint events and actions of the monitored device”). EX1003, ¶160,164.

Finally, as noted above, the monitored events include application events generated while the application is operating. EX1003, ¶¶159-160. For example, the user authentication is part of the context of a launched application. *Id.*

Furthermore, Mukherjee discloses identifying security issues for active objects. EX1003, ¶162. Specifically, Mukherjee states:

The targets of abuse are characterized as being either system objects or user objects and as being either passive or active. User objects are owned by nonprivileged users and/or reside within a non-privileged user's directory hierarchy. System objects are the complement of user objects. Passive objects are files, including executable binaries, ***while active objects are essentially running processes.***

EX1005, p.13. (emphasis added)

Based on this disclosure, a PHOSITA would recognize that the host monitor (“endpoint trust agent”) monitors active objects on the host (“monitored device”) and so monitors running processes for notable events to send to the DIDS director (corresponding to “sending...a dynamic context including endpoint events and actions of applications executing on the monitored device at runtime”). EX1003, ¶¶162-164; EX1003, pp.11-13.

For these reasons, a PHOSTIA would understand that Mukherjee discloses the limitations recited in claim limitation 1[b]. EX1003, ¶165.

1c. receiving, at the trust orchestration server, the dynamic context including the endpoint events of the monitored device and the applications executing on the monitored device at runtime;

As discussed above with respect to limitations 1a and 1b, Mukherjee discloses a DID director (“trust orchestration server”) and discloses transmitting notable events (“dynamic context”) including failed events (“endpoint events”) and rlogin and rsh network access (“actions”) of the monitored host and applications to the DIDS director. EX1003, ¶168.

Mukherjee discloses receiving, at the DIDS director (“trust orchestration server”) information for notable events (“dynamic context”) including failed events (“endpoint events”) related to a monitored device and the applications executing on the monitored device at runtime. EX1003, ¶159.

As discussed above regarding limitation 1[b], the dynamic context is transmitted to the “DIDS director”, which corresponds to the trust orchestration server. EX1003, ¶159,164,168; EX1005, p.13. Mukherjee discloses that the “DIDS director” receives the events and actions that comprise the dynamic context. *Id.*

Specifically, Mukherjee states:

The DIDS director consists of three major components... a communications manager, an expert system, and a user interface... The expert system is responsible for evaluating and reporting on the security state of the monitored system. *It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a*

whole... The director's user interface allows the SSO interactive access to the entire system. The SSO can watch activities on each host, watch network traffic (by setting "wire taps"), and request more specific types of information from the monitors.

EX1005, p.12 (emphasis added).

For these reasons, a PHOSTIA would understand that Mukherjee discloses the limitations recited in claim limitation 1[c]. EX1003, ¶¶169.

1d. analyzing, by the trust orchestration server, the received endpoint events;

Mukherjee discloses analyzing, by the DIDS director ("trust orchestration server"), the notable events ("received endpoint events"). EX1003, ¶¶172-173.

Mukherjee states:

The host and LAN monitors are primarily responsible for the collection of evidence of unauthorized or suspicious activity, while the DIDS director is primarily responsible for its evaluation...

The host monitor consists of a host event generator (HEG) and a host agent. The HEG collects and analyzes audit records from the host's operating system. The audit records are scanned for notable events, which are transactions that are of interest independent of any other records. These include, among others, failed events, user authentications, changes to the security state of the system, and any network access such as rlogin and rsh. These notable events are then sent to the director for further analysis...

It receives the reports from the host and the LAN monitors, and, based on these reports, it makes inferences about the security of each individual host, as well as the system as a whole.

EX1005, p.11-12 (emphasis added).

For these reasons, a PHOSTIA would understand that Mukherjee discloses the limitations recited in claim limitation 1[d]. EX1003, ¶¶174.

1e. receiving, by the trust orchestration server, third party network endpoint assessments;

Mukherjee or Mukherjee in view of Miliefsky discloses receiving, by the DIDS director (“trust orchestration server”), “third party network endpoint assessments” in the form of information or assessments from Network Security Monitoring (NSM) Services. EX1003, ¶178.

Mukherjee discloses that the “DIDS system” receives information from the “NSM services”, and the “NSM services” provide network assessment to clients. *Id.* The NSM itself is an outside monitor of the LAN, and hence a “third party,” and is mostly invisible to the normal flow on the network. EX1005, pp.9-10; EX1003, ¶180. This independent monitor may include a LAN event generator (LEG) that observes and records events on the LAN. Ex1003, ¶181. Specifically, Mukherjee states:

Like the host monitor, the LAN monitor consists of a LAN event generator (LEG) and a LAN agent. The LEG is currently a subset of the *NSM* [17,27, 29]. Its main responsibility is to observe all of the traffic

on its segment of the LAN in order to monitor host-to-host connections, services used, and volume of traffic. The LAN monitor reports on network activities such as rlogin and telnet connections, the use of security-related services, and changes in network traffic patterns.

EX1005, p.12 (emphasis added).

NSM services are a “third party” because NSM servers are remote from the DIDS system and are not directly involved with the transactions between the DIDS director and the host monitor or other computing devices on the network. EX1003, ¶182. Thus, the NSM services are external to the DIDS director and its analysis as well as the rest of the network. *Id.* NSM services are an “endpoint assessment” because the NSM services provide network assessments regarding the hosts or device (“endpoint” connected to the LAN) and such network endpoint assessments are well known entities. EX1003, ¶179,182(citing EX1020); EX1005, p.12.

Further, Mukherjee explicitly endorses transmitting audit records from third-party packages:

The objects at the first (i.e., lowest) level are the audit records provided by the **host OS, by the LAN monitor, or by any third-party auditing package**. The objects at this level are both syntactically and semantically dependent on the source. At this level, all of the activity on the host or LAN is represented.

EX1005, p.13 (emphasis added).

Thus, Mukherjee at least discloses a third-party auditing package which is a package from another vendor that is a “third party network endpoint assessment.” EX1005, p.13; EX1003, ¶182.

Further, Miliefsky discloses a network security apparatus that incorporates data from many components as well as “third party sources.” EX1003, ¶¶184-185. Miliefsky discloses that the security appliance collects a wide range of data including vulnerability assessments from INFOSEC plugins, and these assessments use CVEs, which are third-party vulnerability information, to make the assessments. EX1003, ¶185. Specifically, Miliefsky states:

An Information Security (INFOSEC) software engine acts as a gateway between users, personal computers, servers, services and the computer network (internet, intranet, extranet, wide area network, wireless network or local area network). The information Security (INFOSEC) Engine controls the computer-based network scanning, standards-based vulnerability assessment through common vulnerabilities and exposures (CVEs) testing, reporting and remediation as well as interfacing with the INFOSEC ENGINE PLUGINS. The INFOSEC Engine is structured in a modular fashion with a main controller that takes input for control from the Secure GUI modules. Functions include reading and acting upon the configuration and scheduling data as stored by the Secure GUI modules. ***The INFOSEC Engine contains a unique module for each vulnerability assessment CVE test as well as communication modules to enable non-intrusive testing for each unique IP Address accessible from the computer-based network. The***

INFOSEC Engine contains read, write and export functionality for vulnerabilities found and reported in various formats including but not limited to structured query language (SQL) databases and tables, portable document format (PDF), extensible markup language (XML), hypertext markup language (HTML), comma separated values (CSV) and Excel file format (XSL). The INFOSEC Engine, at initialization, is able to determine which CVE tests are available as well as which INFOSEC Engine Plugins are available and then to relay this information to the Secure GUI for administration, control and management.”

EX1006, ¶[0049] (emphasis added).

Further, Miliefsky discloses that the database, that holds results for this system, can include external third-party sources. EX1003, ¶185-187. Miliefsky states:

Database Correlation and Warehousing Engine: **This engine gathers data from processes and results throughout the product as well as from internal/external resources, including but not limited to the Update Servers, Countermeasure appliances, Data feeds, other devices of the same nature as this patent describes and any related third party sources.** The engine uses data warehouse methodologies to store this data. The engine also provides a means of querying the database and warehouse information either through automated methods or through on-demand user interfaces.

EX1006, ¶[0093] (emphasis added).

Thus, in addition to the external assessments described in Mukherjee, Miliefsky teaches receiving such assessments of a network endpoint from a third-party vendor. EX1006, ¶¶0104]; EX1003, ¶¶188-191. A PHOSITA would have been motivated to combine Miliefsky's network assessments with Mukherjee's IDS system because Miliefsky already describes combining such assessments with an IDS and notes the advantages. EX1003, ¶192. Miliefsky states:

In the event the INFOSEC countermeasure is an IDS, dynamic alerting of the IT manager or an alternative alert recipient and **sharing the related CVE tests data with the IDS to help the IDS reduce false positives in the IDS alerting module as well as reduce the traffic load** related to intrusion detections which attack a particular IP address that is not susceptible to that particular attack methodology based upon the related CVE tests data and will take place through the Countermeasure Communications Client plug-in which has been written specifically for that MAKE, MODEL and VERSION IDS. EX1006, ¶¶0064] (emphasis added).

Thus, Miliefsky teaches why a PHOSITA would be motivated to combine Miliefsky's data into an IDS's operation because: it will reduce false positives and will reduce the traffic load. EX1003, ¶¶192-193. A PHOSITA also would have known that any of Miliefsky's proactive fixes, which prevent an attacker from getting into the system at all, would also reduce the load on the IDS. EX1003, ¶¶117-123,193.

For these reasons, a PHOSTIA would understand that Mukherjee alone or the combination of Mukherjee and Miliefsky discloses the limitations recited in claim limitation 1[e]. EX1003, ¶194.

1f. generating, by the trust orchestration server, temporal events based at least in part on analyzing the third party network endpoint assessments;

Mukherjee in view of Miliefsky discloses generating, by the trust orchestration server (“DIDS director”), temporal events based at least in part on analyzing the third-party network endpoint assessments (“Network Security Monitoring (NSM) Services” as modified by Miliefsky). EX1003, ¶¶187-188,198. As explained above with respect to limitations 1[a] and 1[e], the combination of Mukherjee and Miliefsky discloses a trust orchestration server which receives and processes third party network endpoint assessments. EX1006, ¶¶[0093],[0104]; EX1003, ¶¶147-150,189-190,202.

Regarding the “temporal events” introduced in this limitation, the ’616 patent states that “temporal events...annotate the state of the endpoints (devices) at scan time,” that they can be “expressed as assertions about operational parameters (e.g., vulnerabilities, compliance, patch level, etc.) based on enterprise policies” and can include “trust and severity scores.” EX1003, ¶197; EX1001, 11:40-60. Miliefsky discloses a “network based assets list” that is derived from a network scan to determine the assets and configuration of the network which corresponds to

temporal events because the “network based assets list” can be expressed as assertions about operational parameters. EX1003, ¶¶198-199; EX1006, ¶¶[0040]-[0041],[0103]-[0104]. Miliefsky describes using the same open source framework (OVAL) for this collection of configuration data as described by the ’616 patent. EX1001, 11:59-12:16; EX1003, ¶¶200-201. Accordingly, a PHOSITA would have understood that the OVAL implementation of Miliefsky discloses generating temporal events as claimed. *Id.*

Further, Miliefsky discloses that this configuration data collected or processed at particular time (“temporal events”) can be combined with vulnerability assessment data and include related data, which as explained above, is disclosed as being sourced from third parties. EX1003, ¶¶201. Likewise, internal assessments such as the generated network asset list are disclosed as being combined with external assessments to provide feedback to each other (e.g., to produce control lists). EX1003, ¶¶202-203; EX1006, ¶[0094]. Therefore, Miliefsky teaches “generating...temporal events based at least in part on analyzing the third party network endpoint assessments.” EX1003, ¶¶198-201.

For these reasons, a PHOSITA would understand that the combination of Mukherjee and Miliefsky discloses the limitations recited in claim limitation 1[f]. EX1003, ¶208.

1g. correlating, by the trust orchestration server, the received endpoint events and the generated temporal events; and

As described above regarding claim limitation 1[a], Mukherjee teaches a DIDS director that corresponds to the claimed “trust orchestration server.” EX1003, ¶¶147-151; EX1005, p.11. Likewise, as described regarding claim limitations 1[b] and 1[f] the DIDS director receives a network asset list that corresponds to the “temporal events” and receives device and application behavior data and audit records which correspond to “endpoint events.” EX1003, ¶¶198,201,212; EX1005, p.12.

Mukherjee discloses that the data (e.g., notable events, interesting events, failed events, etc.) from the host and LAN monitors are used by the DIDS director as the basis for inferences about the security of individual hosts. EX1005, p.11-13; EX1003, ¶213. Further, the DIDS director includes a rule-based expert system that incorporates an intrusion detection model (IDM) that detects patterns in the data in each layer of this model. EX1005, pp.12-13. For example, regarding the fourth layer of the IDM, Mukherjee states:

The fourth layer of the model introduces the event in context. There are two kinds of context: temporal and spatial. As an example of temporal context, behavior which is unremarkable during standard working hours may be suspicious during off-hours [35]. The IDM, therefore, allows for the application of information about wall-clock time to the events it is considering. Wall-clock time refers **to information about**

the time of day, weekdays versus weekends and holidays, as well as periods when an increase in activity is expected. In addition to the consideration of external temporal context, **the expert system uses time windows to correlate events occurring in temporal proximity.** This notion of temporal proximity implements the heuristic that a call to the UNIX who command followed closely by a login or logout is more likely to be related to an intrusion than either of those events occurring alone. **Such related patterns of behavior are also referred to as attack signatures** [23,24].

EX1005, p.13 (emphasis added)

In this quoted portion of its disclosure Mukherjee explicitly teaches correlating temporal context based on time windows. EX1005, p.13; EX1003, ¶¶214-216. Further, the events themselves are analyzed for patterns (i.e., relationships between events) based on signatures of an attack. EX1003, ¶¶63-65,214(citing EX1007, EX1008). Additionally, would have been obvious to a PHOSITA to include the temporal events discussed with respect to limitation 1[f] into the correlation operations of Mukherjee as suggested by Mukherjee in the quote above. EX1003, ¶216. Therefore, Mukherjee as modified by Miliefsky (for example, in limitation 1[f]) teaches correlating the elements of this limitation at the DIDS director. EX1003, ¶¶216-219.

For these reasons, a PHOSITA would understand that the combination of Mukherjee and Miliefsky discloses the limitations recited in claim limitation 1[g]. EX1003, ¶220.

1h. generating, by the trust orchestration server, an integrity profile for the system.

As described above regarding claim limitation 1[a], Mukherjee teaches a DIDS director that corresponds to the claimed “trust orchestration server.” EX1003, ¶¶147-151; EX1005, p.11. Further, the DIDS director includes a rule-based expert system that incorporates an intrusion detection model (IDM) that detects patterns in the data in each layer of the model. EX1005, pp.12-13. At the fifth layer of the IDM, the network events (e.g., audit records, behavior) and temporal context combined as described regarding claim 1[g] are analyzed to identify threats. EX1005, p.13; EX1003, ¶198,216. The threats identified include network elements (e.g., system objects) that constitute targets. EX1003, ¶223. A PHOSITA would have recognized this threat detection related to components of the network as an “integrity profile” for the system, consistent with the teachings of the ’616 patent. EX1003, ¶224; EX1001, 12:41-44 (“integrity profile represents an aggregation of system warnings (threats such as malware) identified based on the received temporal and endpoint events”).

Further, Miliefsky also discloses generating an integrity profile of the system. Specifically, Miliefsky discloses “[t]he method defines a true risk profile

for the computer-based network environment, and uses the knowledge of external and internal CVEs as well as how to manage and remediate against these CVEs, to provide more robust and proactive security.” EX1006, ¶¶0045]; EX1003, ¶225. Miliefsky also teaches displaying the risk profiles of assets (e.g., equipment) on administrative dashboards, stating “[t]he administrative dashboard provides access to and control of ...regulatory compliance reporting and related network-based asset risk profile.” EX1006, ¶¶0070]; EX1003, ¶225. A PHOSITA would have recognized a risk profile, generated from vulnerability assessments, to be an “integrity profile” of the system, consistent with the teachings of the ’616 patent. EX1003, ¶225; EX1001, 12:41-44.

For these reasons, a PHOSITA would understand that Mukherjee and Miliefsky each discloses the limitations recited in claim limitation 1[h].

Because Mukherjee in view of Miliefsky discloses every limitation recited in claim 1 as explained above, a PHOSITA would have found claim 1 to be obvious. EX1003, ¶228. Therefore, claim 1 should be found invalid under 35 U.S.C. § 103(a).

2. Claim 2

The method of claim 1, wherein the third party network endpoint assessments are received from an endpoint assessment service.

As described regarding claim limitation 1[e], Mukherjee alone or Mukherjee in view of Miliefsky teaches “third party network endpoint assessments.” EX1003,

¶¶178-193,233; EX1005, p.12-13; EX1006, ¶¶[0094],[0104]. Further, Miliefsky teaches OVAL-compatible vulnerability assessment systems and processes that are hosted as external websites or services. EX1003, ¶¶200,231; EX1006, ¶¶[0104]. A PHOSITA would have recognized that these vulnerability/security services of Miliefsky that are hosted by a third-party website would correspond to “an endpoint assessment service.” EX1003, ¶¶231-232.

Therefore, a PHOSITA would have found the limitations recited in claim 2 to be obvious based on Mukherjee in view of Miliefsky. EX1003, ¶233. As shown, the Mukherjee- Miliefsky combination discloses all limitations of claims 1 from which claim 2 depends. Accordingly, claim 2 should be found invalid as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky.

3. Claim 3

The method of claim 2, further comprising: receiving, by the trust orchestration server, integrity measurement and verification reports from the endpoint assessment service.

As described regarding claim limitation 1[e] and claim 2, Mukherjee in view of Miliefsky teaches “third party network endpoint assessments” and “endpoint assessment services.” EX1003, ¶¶178-193,231,237; EX1005, p.12-13; EX1006, ¶¶[0094],[0104]. Specifically, Miliefsky teaches an OVAL framework for assessing vulnerabilities based on collected third-party information (e.g., CVEs). EX1003, ¶237. Further, Miliefsky discloses receiving schemas, countermeasures,

regulatory rules via a feed as part of the OVAL framework so that it can perform a real-time CVE differential analysis based on the information. EX1006, ¶¶[0107]-[0110]; EX1003, ¶237. This system also performs regulatory compliance reporting, enhanced auditing, and CVE auditing that is based on the OVAL results schema (corresponding to “receiving... verification reports from the endpoint assessment service”). EX1006, ¶¶[0067],[0084],[0103]; EX1003, ¶238. Similarly, Miliefsky discloses automated vulnerability reporting via its reporting system that evaluates systems for vulnerabilities and calculates a risk profile (corresponding to “receiving, by the trust orchestration server, integrity measurement”). EX1006, ¶¶[0055],[0057],[0084]; EX1003, ¶¶99,225. Likewise, the ’616 patent describes “integrity measurement and verification reports” as similar to or alternative to endpoint assessments and involving audit and scanning services. EX1001, 2:24-34,32:63-67; EX1003, ¶¶237-238. A PHOSITA would have recognized that the compliance reporting and CVE auditing of Miliefsky corresponds to the “integrity measurement and verification reports from the endpoint assessment service” received at the server. EX1003, ¶237.

Therefore, a PHOSITA would have found the limitations recited in claim 3 to be obvious based on Mukherjee in view of Miliefsky. EX1003, ¶240. As shown, the Mukherjee- Miliefsky combination discloses all limitations of claims 1 and 2 from which claim 3 depends. Accordingly, claim 3 should be found invalid

as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky.

4. Claim 4

The method of claim 1, wherein the third party network endpoint assessments are received from one or more collaboration services.

As described regarding claim limitation 1[e], Mukherjee in view of Miliefsky teaches “third party network endpoint assessments.” EX1003, ¶¶178-193,231; EX1005, p.12-13; EX1006, ¶¶[0094],[0104]. Further, Miliefsky teaches OVAL-compatible vulnerability assessment systems and processes that are hosted as external websites or services. EX1003, ¶243; EX1006, ¶[0104]. Indeed, claim 4 simply replaces “endpoint assessment service” in claim 2 with “collaboration services.” EX1003, ¶¶242-243. As described regarding claim 2, Miliefsky teaches OVAL compatible tools, databases, and Websites for providing reporting on vulnerabilities and configuration settings. EX1006, ¶[0104]. A PHOSITA would have recognized that the OVAL tools described by Miliefsky correspond to the claimed “collaboration services” because Miliefsky describes the OVAL tools, websites, etc. as resulting from a collaboration of various entities funded by the U.S. Department of Homeland Security. EX1006, ¶¶[0101]-[0107]; EX1003, ¶¶242-243.

Therefore, a PHOSITA would have found the limitations recited in claim 4 to be obvious based on Mukherjee in view of Miliefsky. EX1003, ¶¶245. As

shown, the Mukherjee-Miliefsky combination discloses all limitations of claims 1 from which claim 4 depends. Accordingly, claim 4 should be found invalid as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky.

5. Claim 5

The method of claim 4, further comprising: receiving, by the trust orchestration server, integrity measurement and verification reports from the one or more collaboration services.

As described regarding claim limitations 1[a] and claim 4, Mukherjee teaches “trust orchestration server” and Mukherjee in view of Miliefsky teaches “one or more collaboration services.” EX1003, ¶¶178-193,243; EX1005, pp.12-13; EX1006, ¶¶[0094],[0104]. Likewise, Miliefsky teaches OVAL-compatible vulnerability assessment systems and processes that are hosted as external websites or services. EX1003, ¶243; EX1006, ¶[0104]. Further, Miliefsky discloses receiving schemas, countermeasures, regulatory rules via a feed that can perform a real-time CVE differential analysis based on the information. EX1006, ¶¶[0107]-[0110]; EX1003, ¶¶243,249-250. This system also performs regulatory compliance reporting, enhanced auditing, and CVE auditing. EX1006, ¶[0067]; EX1003, ¶¶250-251. Likewise, the ’616 patent describes “integrity measurement and verification reports” as similar to or alternative to endpoint assessments and involving audit and scanning services. EX1001, 2:24-34,32:63-67; EX1003, ¶236-237,250. A PHOSITA would have recognized that the compliance reporting and CVE auditing of Miliefsky corresponds to the “integrity measurement and

verification reports from one or more collaboration services” received at the server.

EX1003, ¶¶236-237,250-251.

Therefore, a PHOSITA would have found the limitations recited in claim 5 to be obvious based on Mukherjee in view of Miliefsky. EX1003, ¶253. As shown, the Mukherjee-Miliefsky combination discloses all limitations of claims 1 and 4 from which claim 5 depends. Accordingly, claim 5 should be found invalid as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky.

6. Claim 6

6a. The method of claim 1, wherein the endpoint events of the monitored device are received by a system event correlator of the trust orchestration server from endpoint trust agents, and

As described regarding claim limitations 1[a] and 1[b], Mukherjee teaches “endpoint events of the monitored device” and “trust orchestration server.” Mukherjee teaches an intrusion detection model (IDM) that correlates events so as to identify threats based on data from monitors including host monitors on a host (“endpoint trust agent”). EX1003, ¶¶141-145. The DIDS director includes an expert system of rules based logic that correlates system events. EX1003, ¶¶150,256. The different levels of the IDM are part of the expert system because the “DIDS utilizes a rule-based (or production) expert system written in CLIPS. The expert system uses rules derived from a hierarchical Intrusion Detection

Model (IDM).” EX1005, p.12-13; EX1003, ¶150,256. Mukherjee discloses correlating events in the “fourth layer” of the Intrusion Detection Model:

The fourth layer of the model introduces the event in context. There are two kinds of context: temporal and spatial. As an example of temporal context, behavior which is unremarkable during standard working hours may be suspicious during off-hours [35]. The IDM, therefore, allows for the application of information about wall-clock time to the events it is considering. Wall-clock time refers to information about the time of day, weekdays versus weekends and holidays, as well as periods when an increase in activity is expected. In addition to the consideration of external temporal context, the expert system uses time windows to correlate events occurring in temporal proximity. This notion of temporal proximity implements the heuristic that a call to the UNIX who command followed closely by a login or logout is more likely to be related to an intrusion than either of those events occurring alone. Such related patterns of behavior are also referred to as attack signatures [23,24].”

EX1005, p.13 (emphasis added).

A PHOSITA would have recognized the “DIDS director” of Mukherjee as corresponding to the “trust orchestration server” and its IDM logic as corresponding to a “system event correlator.” EX1003, ¶150-151,256; EX1001, 14:25-45.

For these reasons, a PHOSTIA would understand that Mukherjee discloses the limitations recited in claim limitation 6[a]. EX1003, ¶257.

6b. the temporal events are generated by a trust broker of the trust orchestration server and are received by the system event correlator from the trust broker.

As described regarding claim limitation 1[f], Miliefsky teaches generating “temporal events” similar to those disclosed in the ’616 patent. EX1003, ¶¶198-201; EX1001, 11:50-60. Specifically, Miliefsky discloses “in a first step the network is scanned and/or probed for any and all attached equipment and related assets, herein referred to as “network-based” assets” which is used to generate a network asset list (“temporal events”). EX1006, ¶¶[0040]-[0041]; EX1003, ¶¶198-200,260. Likewise, both the ’616 patent and Miliefsky reference using the OVAL framework to accomplish this function. EX1006, ¶¶[0103]-[0104]; EX1003, ¶¶261-263; EX1001, 11:59-12:16. Finally, Miliefsky teaches that the database that stores the collected network asset information can include external third-party assessments as well. EX1003, ¶¶262,264; EX1006, ¶[0093]. Accordingly, the engine managing the database/correlation is a broker for data relating to security and trust of assets on the network (“trust broker”). EX1003, ¶265.

For these reasons, a PHOSTIA would understand that the combination of Mukherjee and Miliefsky discloses the limitations recited in claim limitation 6[b]. EX1003, ¶¶260-266.

Therefore, a PHOSITA would have found the limitations recited in claim 6 to be obvious based on Mukherjee in view of Miliefsky. EX1003, ¶267. As

shown, the Mukherjee- Miliefsky combination discloses all limitations of claims 1 from which claim 6 depends. Accordingly, claim 6 should be found invalid as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky.

C. GROUND 2: Claim 7 is unpatentable under 35 U.S.C. § 103(a) as obvious in view of Mukherjee (EX1005), Miliefsky (EX1006), and Guo (EX1011).

Motivation to Combine Mukherjee, Miliefsky, and Guo

Petitioner evaluates the scope and content of the prior art, any differences between the art and the claim, and the knowledge of person of ordinary skill in the art in accordance with *Graham v. John Deere Co.*, 383 U.S. 1 (1966) and *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398 (2007). Petitioner is unaware of any secondary considerations relevant to the '616 patent and a more than adequate written rationale is provided to support the combination of Mukherjee, Miliefsky, and Guo under *KSR*. 550 U.S. at 418.

As described above regarding the motivation to combine Mukherjee and Miliefsky, each of these references describes interoperability and open platforms as being key to integrating additional functionality and detection capabilities.

EX1003, ¶¶117-123. That is, each of these references anticipate a PHOSITA adding an analyzer, monitor, or scanner beyond those disclosed in their disclosures. EX1006, ¶[0064]; EX1005, p.12; EX1003, ¶¶117-123,270.

Specifically, Mukherjee teaches incident handling tools that are to be combined with DIDS, while Miliefsky teaches incorporating an anti-virus plugin. *Id.* A PHOSITA would have recognized that Mukherjee and Miliefsky teach combinations with processes and components like the anti-virus scanner for memory disclosed in Guo. EX1003, ¶270.

Further, a PHOSITA would have recognized that Guo teaches a real-time defense mechanism that detects malicious code that is dynamically created (revealed) even if the code is created at runtime (e.g., by unpacking it). EX1011, 2:20-30; EX1003, ¶¶69-72, 271(citing EX1018, EX1019). Indeed, Guo enables dynamically-created code to be unpacked and then scanned for viruses before execution; thus, allowing response in real time. *Id.* Guo explains a “packer is a tool that is used to take an existing piece of malware and hide or ‘pack’ it to make it no longer detectable to most signature-based detection systems....Because signatures used in anti-virus (AV) scanning engines are derived from the byte-level representations of malware samples, malware writers use packers to hide the malware program by changing its appearance. Even worse, malware writers can also apply different packers in different combinations to create a large number of variants of existing malware that can easily evade signature-based AV Scanners.” EX1011, 1:35-43; EX1003, ¶274. A PHOSITA would recognize that the scanners

of Mukherjee and Miliefsky may be evaded in this manner and would look to Guo to improve virus detection in the packed code of executables. EX1003, ¶¶274-276.

Additionally, as described in Guo, its process provides a more efficient and harder to evade method of extracting and scanning packed code than earlier attempts at manual reverse engineering the packers. EX1011, 1:43-2:35,3:24-54; EX1003, ¶¶110,115,276. By using the unpackers provided with the executable and scanning the unpacked code in real time in memory, the system of Guo is also able to reverse the packing sequence and scan the packed code before execution for different packers and multiple-packed code. EX1011, 5:17-55; EX1003, ¶¶274-276. Therefore, a PHOSITA would have looked to Guo as the better solution to the problem of detecting viruses in packed code, which can be malicious or benign. EX1003, ¶¶274-276.

Thus, A POSITA would have been motivated to combine Guo with Mukherjee and Miliefsky, and would have had a high expectation of success with minimal experimentation. EX1003, ¶¶273,277.

1. Claim 7

The method of claim 1, further comprising: performing, by a malware analyzer, detection of code obfuscation and evasion techniques in reverse engineered code of the applications, the code obfuscation and evasion techniques including at least one of: multiple packing, anti-debug, anti-trace, anti-memory, and anti-emulation.

As described in Ground 1, the combination of Mukherjee and Miliefsky teaches all the features of claim 1 from which claim 7 depends.

Guo teaches the additional limitations of claim 7 which is directed to a malware analyzer. Guo discloses a malicious code detection module together with a packing manager which corresponds to the claimed malware analyzer. EX1003, ¶280; EX1011, 5:17-55, Fig.-3. The packing manager, which includes the malicious code detection module, scans code that has been unpacked. EX1003, ¶281-283; EX1011, 5:17-55. Guo notes that the “unpacking” can “include unpacking, unencrypting, descrambling, deobfuscating, etc.” *Id.* Further, Guo directly addresses detection evasion techniques that are solved by the disclosed combination of scanner and packing manager. EX1011, 11:47-12:27. Therefore, a PHOSITA would recognize that Guo teaches “performing, by a malware analyzer, detection of code obfuscation and evasion techniques in ...code of the applications.” EX1003, ¶283-284.

Further, Guo explains that the disclosed malicious code detection module operates to analyze memory pages where code has been unpacked by the unpacker bundled with the executable. EX1011, 9:15-65; EX1003, ¶286. Guo notes that for

multiple layers of packed code and unpackers (multiple packed code), the process operates in the same manner and can analyze the code for viruses before the virus can infect the computer. EX1011, 5:17-55,9:15-65. Therefore, Guo teaches detecting “the code obfuscation and evasion techniques including at least one of: multiple packing...” of claim 7. EX1003, ¶¶281-285.

Finally, Guo teaches “detection ...in reverse engineered code of the applications” as construed above. EX1003, ¶288. Specifically, Guo discloses detecting “malicious code that is dynamically created (i.e., created at runtime), e.g., by a packer, antivirus software may require dynamically created code to be scanned before it can be executed by a host computer system.” EX1011, 2:20-25; EX1003, ¶¶281-285,288. Guo detects characteristics about the dynamically created code unpacked from the executable application such as extracting the original code for analysis with an anti-virus scanner. EX1003, ¶287. Thus, a PHOSITA would have understood Guo’s disclosures, which automatically extract the original malware code from evasive packing to be “reverse engineering” the code and, therefore, teaches this limitation. EX1003, ¶¶285-288.

Therefore, a PHOSITA would have found the limitations recited in claim 7 to be obvious based on Mukherjee in view of Miliefsky and Guo. EX1003, ¶¶289-290. As shown, the Mukherjee-Miliefsky-Guo combination discloses all limitations of claims 1 from which claim 7 depends. Accordingly, claim 7 should

be found invalid as unpatentable under 35 U.S.C. § 103(a) based on Mukherjee in view of Miliefsky and Guo.

VI. NO BASIS FOR DISCRETIONARY DENIAL

There is no basis to deny institution under 35 U.S.C. § 325(d).

There is parallel court litigation, so *Fintiv* factors are addressed preliminarily. First, no stay has been entered in the parallel litigation. Second, trial is currently set to begin February 5, 2024. Third, it is early in the parallel litigation as neither party has responded to written fact discovery nor taken any other type of discovery, nor have the parties begun the Markman process. Fourth, Defendants in the parallel litigation served preliminary invalidity contentions on March 28, 2023. Fifth, Petitioner and Taasera are parties in the parallel litigation as identified in the Mandatory Notices below and the Challenged Claims are the same as the claims identified in Taasera infringement contentions. As such, the first five *Fintiv* factors do not favor discretionary denial. Moreover, even if that were not the case, the merits of the grounds raised in this Petition are particularly strong for all of the reasons discussed above.

Discretionary denial is strongly disfavored here because the “same or substantially the same art or arguments” were not presented to the Office, and the Office “erred in a manner material” to patentability by not asserting the Mukherjee,

Miliefsky, and Guo references. *See Advanced Bionics, LLC v. MED-EL*

Elektromedizinische Gerate GmbH, IPR2019-01469, Paper 6, 8-9 (Feb. 13, 2020).

Analysis of this issue is guided by the “*Becton*” factors. *Id.*, 9-11.

Mukherjee, Miliefsky, and Guo were not cited, individually or in combination, in any of the Office communications. *See* EX1006. Regarding *Becton* factors (a) (material differences between asserted art and prior art) and (b) (cumulative nature of the asserted art and prior art), Mukherjee, Miliefsky, and Guo are materially different from the combination of references asserted in Office communications because Mukherjee, Miliefsky, and Guo disclose the independent claim limitations that sending, by the endpoint trust agent on a monitored device, a dynamic context including endpoint events and actions of the monitored device and applications executing on the monitored device at runtime. Thus, the individual references of Mukherjee, Miliefsky, and Guo are not merely cumulative of references the Office relied on. Regarding *Becton* factor (d) (extent of overlap), the arguments in this Petition do not overlap with arguments made during examination.

Even if the Board finds that the “same or substantially the same” art or arguments were presented to the Office, discretionary denial under § 325(d) is still not proper because the Office made a “material error” in not considering the Mukherjee, Miliefsky, and Guo references.

Regarding *Becton* factors (c) (evaluation of art during examination), none of Mukherjee, Miliefsky, and Guo were used in any claim rejections, weighing strongly against exercising discretion. *See Cellco Partnership v. Huawei Device Co.*, IPR2020-01117, Paper 10, 13 (Feb. 3, 2021).

Regarding *Becton* factor (e) (examiner error) and (f) (other facts), this petition specifically points out and explains how Mukherjee, Miliefsky, and Guo discloses all limitations of the claims.

VII. CONCLUSION

Inter partes review of claims 1-6 and 7 of U.S. Patent No. 9,092,616 is requested based on the grounds described above.

VIII. Mandatory Notices (37 C.F.R. § 42.8(a)(1))

REAL PARTY IN INTEREST:

The real parties-in-interest are Trend Micro, Inc., Trend Micro America, Inc. and Trend Micro Incorporated. Trend Micro, Inc. is a wholly owned subsidiary of Trend Micro America, Inc., which is a wholly owned subsidiary of Trend Micro Incorporated, a Japanese Corporation.

RELATED MATTERS:

Pursuant to 37 C.F.R. § 42.8(b)(2), Petitioner identifies the following related matters.

A petition for *inter partes* review is being filed concurrently with respect to U.S. Patent No. 8,990,948, which is related to the '616 patent through a common claim to the same provisional Application No. 61/641,007.

A petition for *inter partes* review has been filed by Palo Alto Networks, Inc as IPR2023-00705 with respect to U.S. Patent No. 8,990,948, which is related to the '616 patent through a common claim to the same provisional Application No. 61/641,007.

A petition for *inter partes* review has been filed by Palo Alto Networks, Inc as IPR2023-00706 with respect to the '616 patent.

The '616 patent is also at issue in the following district court litigations:

- *In re: Taasera Licensing LLC, Patent Litigation*, 2:22-md-03042 (E.D. Tex.), which was consolidated by the Judicial Panel on Multidistrict Litigation on August 3, 2022.

LEAD AND BACKUP COUNSEL:

Pursuant to 37 C.F.R. § 42.8(b)(3) and 37 C.F.R. § 42.10(a), Petitioner appoints:

Robert M. Hansen (Reg. No. 43,656) as lead counsel,

Matthew Anderson (Reg. No. 78,257) as back-up counsel, and

Scott B. Amankwatia (Reg. No. 78,226) as back-up counsel.

Pursuant to 37 C.F.R. § 42.10(b), a power of attorney is being filed with this designation of counsel.

SERVICE INFORMATION:

Petitioner provides the following service information for designated counsel:

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: (703)-391-2900
Fax: (703)-391-2900

Petitioner consents to electronic service by email to the email addresses:

rhansen@marburylaw.com;
manderson@marburylaw.com;
samankwatia@marburylaw.com; and
patdocketing@marburylaw.com.

PAYMENT OF FEES

The PTO is authorized to charge any fees due during this proceeding to Deposit Account No. 18-1579.

*Petition for Inter Partes Review of
U.S. Patent No. 9,092,616*

Respectfully submitted,

The Marbury Law Group, PLLC

A handwritten signature in black ink, appearing to read "Robert M. Hansen", written over a horizontal line.

Robert M. Hansen, Reg. No. 43,656

Scott B. Amankwatia, Reg. No. 78,257

Attorneys for Petitioner

Date: April 5, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: 703-391-2900
Fax: 703-392-2901

CERTIFICATION OF SERVICE (37 C.F.R. §§ 42.6(e), 42.105(a))

The undersigned hereby certifies that the above-captioned **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 9,092,616**, Petitioner's power of attorney, this petition for *inter partes* review, and all associated exhibits were served in their entireties on April 5, 2023, on the following party via Express Mail: to the Patent Owner by serving to the correspondence address of record for the '616 patent:

Buchanan, Ingersoll & Rooney PC

1737 King Street

Suite 500

Alexandria, VA 22314


Robert M. Hansen, Reg. No. 43,656

Attorney for Petitioner

Date: April 5, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: 703-391-2900
Fax: 703-392-2901

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION

Pursuant to 37 C.F.R. § 42.24 *et seq.*, the undersigned certifies that this Petition complies with the 14,000-word type-volume limitation. The Petition contains 12,668 words, excluding the table of contents, table of authorities, Table of Exhibits, mandatory notices under § 42.8, the signature block, the claims appendix, certificate of service, certificate of word count, appendix of exhibits, and words that appear in figures copied from the cited prior art.

The Marbury Law Group, PLLC

A handwritten signature in black ink, reading "Robert M. Hansen", written over a horizontal line.

Robert M. Hansen, Reg. 43,656
Attorney for Petitioner

Date: April 5, 2023

The Marbury Law Group, PLLC
11800 Sunrise Valley Drive
15th Floor
Reston, VA 20191
Tel: 703-391-2900
Fax: 703-392-2901